Using NFS on the DOMAIN Network

Order No. 010414
Revision 00

Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824

Contents

Chapter 1 Overview

The DOMAIN NFS Prerequisites 1-2
The DOMAIN NFS Components 1-3
How DOMAIN NFS Software Works 1-4

Chapter 2 Enabling Access to Remote File Systems

Mounting a Remote File System 2-2
The -o soft Option 2-3
Mounting a File System in the Root Directory 2-3
With the Naming Server Helper Running on the Network 2-4
Without the Naming Server Helper Running on the Network 2-4
Naming Conventions 2-6
Mounting the Root Directory from a Foreign System .. 2-6
Dismounting a Remote File System 2-7
Dismounting a File System Mounted in the Root Directory 2-8
Displaying Status Information 2-9
Specifying Files on Remote File Systems 2-10
Wildcards and Relative Pathnames 2-10
Special Symbols in Links 2-11
Typed Files 2-11
Extended Pathnames 2-13
Super-User Limitation 2-13

Chapter 3 Accessing Files with Commands

Using the Display Manager 3-1
Using DOMAIN/IX Commands 3-2

Using AEGIS Shell Commands 3-2
Special Characters in Command Lines 3-3
AEGIS Shell Command Support 3-3

Chapter 4 Accessing Files from Your Programs

DOMAIN C Library Functions and
DOMAIN/IX System Calls 4-2
DOMAIN C and DOMAIN/IX Call Restriction 4-7
AEGIS System Calls 4-7
AEGIS Call Restrictions 4-1
ios_$create Restrictions 4-1
ios_$set_conn_flag Restrictions 4-1
ios_$set_obj_flag Restrictions 4-1
ios_$set_rec_type Restrictions 4-1
stream_$redefine Restrictions 4-1
Standard I/O Procedures 4-1
DOMAIN Pascal I/O Procedures 4-1
DOMAIN FORTRAN I/O Statements 4-1
Programmer's Notes 4-1

Appendix A Descriptions of Commands, Servers, and Files

Appendix B Managing NFS on a DOMAIN Network

Arranging Servers on the DOMAIN Network B-1
Single DOMAIN NFS Server B-3
Multiple DOMAIN NFS Servers B-5
Creating Uniform User and Group IDs B-6
Procedure for Category A Sites B-9
Procedure for Category B Sites B-1
Procedure for Category C Sites B-1
Adding New Users and Groups B-1

Appendix C Troubleshooting Guide

Appendix D Error Messages

Index

Illustrations

Figure		Page
1-1	Foreign Host with Multiple File Systems	1-6
B-1	Sample Server Arrangement	B-2
B-2	Single DOMAIN NFS Server Configuration	B-4
B-3	Multiple DOMAIN NFS Server Hosts	B-6
B-4	Categories of Sites Requiring DOMAIN NFS Software	B-8
B-5	Category A Site	B-9
B-6	Category B Site	B-11
B-7	Category C Site	B-12

Procedures

Procedure		Page
B-1	Establishing Uniform User and Group IDs at a Category A Site	B-9
B-2	Establishing Uniform User and Group IDs at a Category B Site	B-11
B-3	Establishing Uniform User and Group IDs at a Category C Site	B-13
B-4	Adding a New User to the DOMAIN and Foreign Networks	B-14
B-5	Adding a New Group to the DOMAIN and Foreign Networks	B-15

Tables

Table		Page
3–1	Restricted AEGIS Shell Commands	3–3
3–2	Unsupported AEGIS Shell Commands	3–4
3–3	AEGIS Shell Commands Requiring DOMAIN Objects	3–5
4–1	C Function and DOMAIN/IX Call Support	4–2
4–2	AEGIS System Call Support	4–8
4–3	DOMAIN NFS Object Types and ios_$create	4–11
4–4	ios_$ Equivalents for name_$ Calls	4–14
C–1	Possible Problems at a DOMAIN Host	C–2
C–2	Possible Problems at a Foreign Host	C–3

Chapter 1

Overview

The DOMAIN implementation of the NFS (Network File System) protocol allows users of DOMAIN workstations and other systems running NFS software to transparently access each other's files. In this document, we use the term "DOMAIN NFS" to distinguish the DOMAIN NFS product from other implementations of the NFS protocol.

The DOMAIN NFS implementation derives several advantages from its close ties with the DOMAIN/IX and AEGIS operating systems, advantages not found in other NFS implementations.

For example, the DOMAIN NFS software lets users at **foreign systems** (i.e., systems other than DOMAIN systems) mount the

DOMAIN network root directory (//) to gain access to file systems anywhere on the DOMAIN network. Users at DOMAIN workstations can similarly mount foreign file systems in the DOMAIN network root directory, thereby enabling users throughout a DOMAIN network to use the same names when accessing those file systems. Other NFS implementations do not recognize the concept of a network-wide root directory; they require a system to explicitly mount a **remote file system** before accessing its contents. (A remote file system is one that belongs to another system.)

Many NFS implementations also require you to remount remote file systems whenever you reboot a system. With the DOMAIN NFS implementation, remote file systems remain mounted until you explicitly unmount them.

Many NFS implementations use a directory as a **mount point** for a remote file system. (A mount point is where a mounted file system joins another file system.) If the directory contains any objects at the time of the mount, the objects become unavailable and remain so until someone unmounts the remote file system and frees the directory. The DOMAIN NFS software uses a dedicated **gateway object** as a mount point (a gateway object is a special file that points to a remote file system), which means that your directories remain accessible.

The DOMAIN NFS Prerequisites

Before you can install the DOMAIN NFS software, you must have the following hardware and software:

- A DOMAIN gateway node connecting your site's DOMAIN network to a foreign network (i.e., any network other than a DOMAIN network) supporting one or more foreign systems
- DOMAIN Software Release 9.5 (or a later version) installed on your workstation
- DOMAIN/IX 9.5 (or a later version) installed on your workstation
- TCP/IP 3.0 (or a later version) installed on your workstation

When you are ready to install DOMAIN NFS software on a workstation, consult the *DOMAIN NFS Software Release Document* for directions. (Consult the Preface of this manual for a list of related manuals and their order numbers.) System administrators should also read Appendix B of this manual for information about arranging servers on the DOMAIN network and creating uniform user and group IDs.

The DOMAIN NFS Components

In the remainder of this document, we refer to DOMAIN workstations and foreign systems running NFS software as **hosts**. A **client host** uses resources provided by a **server host**.

DOMAIN NFS software consists of server daemons, type managers, utility commands, and online **man** pages describing the commands.

The server daemons **mountd**, **nfsd**, and **portmap** (often referred to collectively as "the NFS server") run on a server host and thereby enable client hosts to mount its file systems. To be a server host, a DOMAIN workstation must run all three server daemons.

The type managers **nfs_dir**, **nfs_file**, and **nfs_gate** (often referred to collectively as "the NFS type manager") reside in the /sys/mgrs directory of every DOMAIN NFS host. Type managers intercept requests for specific types of objects and determine under what circumstances those objects can be accessed; for more about type managers, see the Typed Files section later in this chapter. The type managers **nfs_dir** and **nfs_file** regulate access to remote directories and files, respectively; the type manager **nfs_gate** regulates access to DOMAIN NFS gateway objects. When accessed from a DOMAIN NFS host, gateway objects and objects on foreign hosts have the following type identifiers:

gateway object	*nfs_gate*
remote directory	*nfs_dir*
remote file	*nfs_file*

The following NFS utility commands reside in a host's /etc directory:

mount	mount a file system or directory

umount	unmount a file system or directory
showmount	display all remote mounts
nfsstat	display NFS statistics
rpcinfo	display NFS–related information

Appendix A in this manual describes these commands, plus the NFS server daemons and reference files.

To read the online **man** pages describing the commands, servers, and files, use the **man** command from within a DOMAIN/IX Shell process window.

How DOMAIN NFS Software Works

All NFS implementations regard a host as being a client, a server, or both. A client host can mount a file system or a directory located on a server host; thereafter, the client host can access the mounted file system or directory as if it were a local file system or directory. In the DOMAIN NFS implementation, a mounted file system or directory remains mounted until it's explicitly dismounted. (Hereafter, all references to file systems also include directories unless explicitly stated to be otherwise.)

The **/etc/mount** and **/etc/umount** commands mount and dismount a remote file system, respectively. In the DOMAIN NFS implementation, the /etc/mount command actually creates a gateway object that points to the file system. For example, the command line:

/etc/mount -o soft sys1:/usr2/smith/examples /examples

creates the gateway object /examples on the client host executing the command. The gateway object `/examples` points to the file system /usr2/smith/examples on the server host `sys1` until you dismount the remote file system with the **/etc/umount** command.

To access an object within a mounted file system, simply include the name of the file system's gateway object in the pathname of the object. For example, assuming that you executed the **/etc/mount** command in the preceding paragraph, you can now use the pathname /examples/script to access /usr2/smith/examples/script on host `sys1`.

Similarly, you can set your working directory using the gateway object as follows:

cd /examples

A gateway object provides access to objects within its own file system; access to objects in other file systems requires additional gateway objects. For example, the slash (/) directory of a DOMAIN NFS host constitutes a single file system; therefore, mounting the slash (/) directory of that host provides access to all of its objects.

However, as illustrated in Figure 1–1, mounting the slash (/) directory of other types of hosts may provide access to only some of the objects on those hosts. This is because users at other types of hosts sometimes divide the host's directory structure into multiple file systems like those shown in the figure. For example, to access all of the objects on the host in the figure, you must mount all of its file systems; mounting just the slash (/) file system does not provide access to the separate /x and /y file systems. This also means that you must use multiple gateway objects to access all of the host's objects.

Compare the mounting of the multiple file systems of a foreign host with the simplicity of mounting the DOMAIN network. Because of the DOMAIN network root directory (//), users at foreign systems can mount an entire DOMAIN network as if it were a single file system. For example, a user at a foreign system could invoke the command:

/etc/mount -o hard boston:// /apollo

to mount the DOMAIN network root directory as /apollo; in this example, the DOMAIN host boston serves as the intermediary between the foreign system and the DOMAIN network root directory. Should the user at the foreign system then want to access a DOMAIN host such as the workstation //charlie, the user could use the pathname:

/apollo/charlie

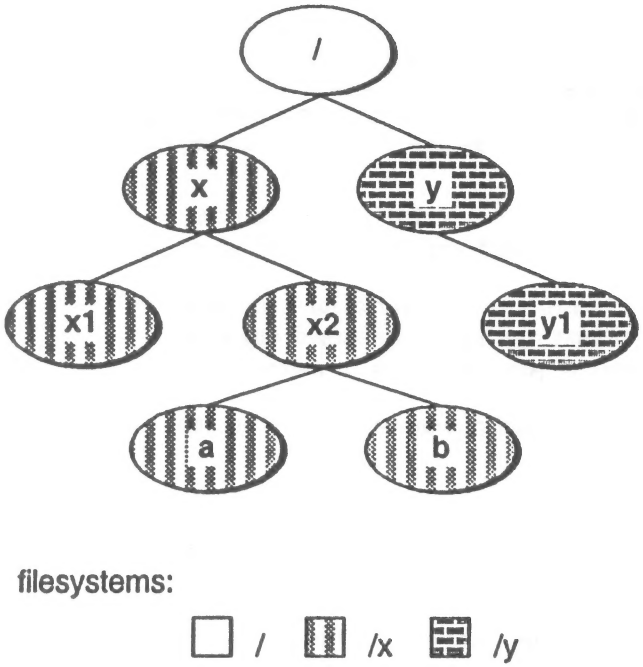

Figure 1–1. Foreign Host with Multiple File Systems

Chapter 2

Enabling Access to Remote File Systems

A DOMAIN NFS host can mount and access file systems on any other NFS host. However, users of DOMAIN workstations generally have no reason to use NFS software to access other DOMAIN workstations since the network–wide DOMAIN file system already provides more capabilities than those available with the NFS software.

The following sections describe how to mount and dismount file systems, access directories and files, and request NFS statistics. To execute the commands described in this chapter, you or your system administrator must first follow the procedures describe in Appendix B

and the *DOMAIN NFS Software Release Document*. For complete descriptions of the commands discussed in this chapter, see Appendix A of this manual.

Mounting a Remote File System

The **/etc/mount** command mounts a remote file system located on a server host. However, only the hosts authorized by the server's /etc/exports file can mount its file systems as clients. For more information about the /etc/exports file, see Appendix A.

NOTE: The command also mounts removable file systems located on disk, diskette, cartridge tape, or magtape. Consult Appendix A or the command's **man** page for directions on mounting removable file systems.

When mounting a remote file system, use the **/etc/mount** command in the form:

/etc/mount [*options*] [*host:fsname*] [*gwname*]

where:

options	is one or more command options.
host:fsname	is the name of the server host, a colon (:), and the pathname of the file system to be mounted; for example, host1:/usr.
gwname	is the pathname of the gateway object; for example, /host1_usr.

If you execute the **/etc/mount** command without arguments, it displays every remote file system currently mounted by the host; the command retrieves this information from the list of mounted file systems it maintains in the /etc/mtab file. For example:

```
/etc/mount
geod:/palm/misc on /geod_misc type nfs (rw)
clue:/usr2/macbeth on /clue_macbeth type nfs (rw)
turban:/usr2/u/gerry on /turban_gerry type nfs (rw)
```

If you specify *host:fsname* and *gwname*, the command mounts the file system *host:fsname* and creates a gateway object named *gwname* that points to the file system. For example, the command:

```
/etc/mount -o soft thor:/usr2 /thor_usr2
```

mounts the file system /usr2 located on the server host thor and creates the gateway object /thor_usr2.

For a complete description of the **/etc/mount** command, consult Appendix A or the command's **man** page.

The -o soft Option

We included the **-o soft** option in the last few sample **/etc/mount** commands for a particular reason. The command defaults to the **-o hard** option for compatibility with other NFS implementations, which means that the command tries repeatedly to mount the specified file system. If a network failure or some other condition makes the file system unavailable, the command ignores the error and continues trying to mount the file system. With the **-o soft** option, the command reports the error and aborts.

For this reason, we recommend that you specify the **-o soft** option with the **/etc/mount** command when mounting a foreign file system from a DOMAIN NFS host. However, if mounting a DOMAIN file system from a foreign system, you should specify the **-o hard** option.

Mounting a File System in the Root Directory

Up to now, we've mounted remote file systems in the slash (/) directory of the client host; for example, /thor_usr2. Actually, you can mount a remote file system in any directory to which you have access, including the network root directory (//).

Mounting a remote file system in the network root directory allows NFS hosts anywhere in the DOMAIN network to use the same path-

name when accessing the file system. For example, mounting the remote file system thor:/usr2 in the network root directory as //thor_usr2 enables all NFS hosts in the DOMAIN network to access the file system using the pathname //thor_usr2.

With the Naming Server Helper Running on the Network

If the DOMAIN network is running the naming server helper NS_HELPER (described in *Administering Your DOMAIN System*), use the **/etc/mount** command with its **-o root** option to mount a remote file system in the network root directory. Because NS_HELPER manages the global network root directory, it automatically makes the file system's name available to all NFS hosts on the network. For example, the command:

```
/etc/mount -o soft,root thor:/usr //thor_usr
```

enables all NFS hosts on the DOMAIN network to access the remote file system thor:/usr with the pathname //thor_usr. Note that you separate two or more options with commas and that the *gwname* argument must begin with double slashes (//).

When you use the **-o root** option, the **/etc/mount** command creates a gateway object in the local host's 'node_data directory. The name of the gateway object consists of the command's *gwname* argument (without the double slashes) and the suffix .nfs_root; for example, thor_usr.nfs_root.

Without the Naming Server Helper Running on the Network

If the DOMAIN network is not running the naming server helper NS_HELPER, then mounting a remote file system in the network root directory requires more of an effort. In this case, the system administrator or the NFS users themselves must execute the same **/etc/mount** command on all NFS hosts in the DOMAIN network. This "manual" approach mounts the remote file system in the local copy of the network root directory located on every DOMAIN NFS host.

To mount a remote file system in the copy of the network root directory on your NFS host, execute the **/etc/mount** command without the **-o root** option but with double slashes (//) before the *gwname* argument. For example, the command:

```
/etc/mount -o soft thor:/usr //thor_usr
```

makes the remote file system thor:/usr accessible to your NFS host as //thor_usr. Executing this command on all NFS hosts in the DOMAIN network has the same effect as using the **-o root** option with the naming server NS_HELPER running on the network.

When you mount a remote file system in a copy of the network root directory, the **/etc/mount** command also creates a gateway object in the local host's 'node_data directory. In this case, the name of the gateway object consists of the command's *gwname* argument (without the double slashes) and the suffix .nfs; for example, thor_usr.nfs.

NOTE: You can explicitly mount a remote file system in your local copy of the network root directory even when NS_HELPER is running on the DOMAIN network. However, we strongly discourage this approach. Since the local copy serves as a cache of the global root directory when NS_HELPER is running, mounting a remote file system in the local copy can cause confusing and incorrect behavior.

As an alternative to executing the same **/etc/mount** command on all NFS hosts in the DOMAIN network, you can execute the command just once and have the NFS users on the network share the resulting mount point. For this approach to work, do not mount the remote file system in a copy of the network root directory; instead, create a mount point (i.e., a gateway object) in a directory that everyone can access. For example, you could use the following command to mount the remote file system thor:/usr and create the gateway object thor_usr in the directory //zipper/gateways:

```
/etc/mount -o soft thor:/usr //zipper/gateways/thor_usr
```

Users at other NFS hosts in the DOMAIN network could then access the remote file system thor:/usr by specifying the pathname:

```
//zipper/gateways/thor_usr
```

Naming Conventions

Whether you mount remote file systems in the network root directory with the **-o root** option or by the manual method, the result is the same: all NFS hosts on the DOMAIN network share the same names when accessing the file systems. System adminstrators therefore need to develop naming conventions to prevent confusion and naming conflicts. We use the following naming convention when specifying the gwname argument of the **/etc/mount** command:

//host_filesystem

where *host* is the host name and *filesystem* is the file system name. For example, we use the *gwname* argument //thor_usr when mounting the remote file system thor:/usr. When mounting a remote file system such as thor:/, we simply use the *gwname* argument //thor.

Mounting the Root Directory from a Foreign System

With most NFS implementations, users who want to access file systems on other NFS hosts must normally execute at least one **/etc/mount** command per host. However, an NFS user at a foreign system need only mount the DOMAIN network root directory (//) once to gain access to file systems located anywhere in the DOMAIN network.

To mount the DOMAIN network root directory from a foreign system, an NFS user selects an NFS server host on the DOMAIN network and executes the **/etc/mount** command in the form:

/etc/mount -o soft host:// directory

where host is the name of the NFS server host and directory is an existing directory on the foreign system. For example, suppose the user selects the NFS server host landfall on the DOMAIN network, creates the directory /apollo, and executes the command:

/etc/mount -o soft landfall:// /apollo

The command transforms the directory /apollo into a mount point for the network root directory. This means that the user at the foreign system can now access a file on any DOMAIN workstation in the network, simply by substituting "/apollo" for the double slashes (//) at the beginning of the file's pathname. For example, the user at the foreign system can now access the DOMAIN file

//bigelow/billy/carousel/barking_sched

with the pathname

/apollo/bigelow/billy/carousel/barking_sched

Dismounting a Remote File System

The **/etc/umount** command dismounts a remote file system mounted on a DOMAIN/NFS host.

NOTE: The command also dismounts removable file systems located on disk, diskette, cartridge tape, or magtape. Consult Appendix A or the command's **man** page for directions on dismounting removable file systems.

The **/etc/umount** command takes the form:

/etc/umount [*options*] { *host:fsname* | *gwname* }

where:

options	is one or more command options.
host:fsname	is the name of the server host, a colon (:), and the pathname of the mounted file system; for example, host1:/usr.
gwname	is the pathname of the gateway object; for example, /host1_usr.

When you execute the command with the *gwname* argument, the command searches the contents of the file /etc/mtab and deletes the first entry it encounters containing the *gwname* argument. For example, the following command dismounts the file system /usr on foreign host thor.

```
/etc/umount /thor_usr
```

The **/etc/umount** command also deletes the gateway object associated with the remote file system.

For a complete description of the **/etc/umount** command, consult Appendix A or the command's **man** page.

Dismounting a File System Mounted in the Root Directory

You can dismount a remote file system mounted in the network root directory in one of two ways, depending on whether the DOMAIN network is running the naming server NS_HELPER.

If the DOMAIN network is running NS_HELPER, dismount the file system using the **/etc/umount** command with its **-root** option. The **-root** option removes the file system's name from the network root directory and deletes the corresponding gateway object (for example, 'node_data/thor_usr.nfs_root). Note that you must execute this command on the NFS host that originally mounted the remote file system; if you attempt to execute the command on another NFS host in the network as in the following example, the command directs you to execute the command at the indicated NFS host:

```
/etc/umount -root //thor_usr
umount: umount -root //thor_usr must be made from node
//tendril
```

If the DOMAIN network is not running NS_HELPER, then use the **/etc/umount** command without the **-root** option: by default, the command dismounts the file system from the local copy of the network root directory. To dismount the remote file system from all copies of the network root directory, the system administrator or the NFS users themselves must execute the command on every NFS host in the DOMAIN network. For example, invoking the following command on every NFS host in the DOMAIN network:

```
/etc/umount //thor_usr
```

dismounts the remote file system mounted as //thor_usr. The command also deletes the gateway object 'node_data/thor_usr.nfs from all NFS hosts that execute the command.

Displaying Status Information

Besides displaying the remote file systems mounted on the local NFS host with the **/etc/mount** command, you can also display NFS statistics with the **/etc/nfsstat** and **/etc/rpcinfo** commands (system adminstrators generally use these commands when administering the network). For complete descriptions of these commands, consult Appendix A. In addition, you can list the client hosts that have mounted file systems of a particular server host with the **/etc/showmount** command.

The **/etc/showmount** command lists the directories of a server host that are mounted by client hosts, displays all remote mounts in the form host:fsname, and lists all exported file systems (i.e., file systems listed in the /etc/exports file of the server host).

The command takes the form:

/etc/showmount [*options*] [*server_host*]

Consult the description of the **/etc/showmount** command in Appendix A, or the command's **man** page, for an explanation of all the available options.

By default, the command lists the client hosts that have mounted file systems located on the local DOMAIN NFS server host. You can also specify a different server host. For example, the command:

```
/etc/showmount thor
```

displays a list similar in form to the sample below:

```
/etc/showmount thor
tripper:
lapis:
karma:
```

Specifying Files on Remote File Systems

Wildcards and relative pathnames work regardless of whether you're specifying files located on a DOMAIN workstation or on a foreign system running NFS software. However, a user at a foreign system cannot use certain extended pathnames or pathnames containing environment variables when accessing objects in DOMAIN file systems. These features are unique to the DOMAIN/IX and AEGIS operating systems and are not recognized by foreign operating systems.

Wildcards and Relative Pathnames

You can use wildcards when specifying pathnames of objects in remote file systems. Simply follow the standard wildcarding procedures that apply to the type of shell in use.

For example, you can use the following AEGIS Shell command to list all objects starting with the characters `test` in the remote directory //thor/results:

```
$ ld //thor/results/test?*
```

Relative pathnames also apply to remote files and directories. For example, you can set your working directory to the remote directory //thor/results and thereafter specify individual files by their leafnames.

```
$ wd //thor/results
$ ld test?*
```



```
Directory  "//terminus/sys/node_data/thor.nfs_root":

test1   test2   test3   test4

4 entries listed.
$
```

Special Symbols in Links

Links in the DOMAIN/IX and AEGIS operating systems can contain unique symbols and environment variables in their text. For example, the following links include the `node_data and - symbols and the DOMAIN/IX $(systype) environment variable in their text:

```
memos                   "-memos"
/etc                    "$(systype)/etc"
/etc/exports            "`node_data/etc.exports"
```

Foreign operating systems cannot translate pathnames containing such links because they do not know how to interpret these unique symbols. You should therefore omit the symbols and substitute their equivalent text; for example, you can substitute the complete pathname /sys/node_data/etc.exports for the link /etc/exports, the pathname /bsd4.2/etc or /sys5/etc for the link /etc, and the pathname *your_naming_directory*/memos for the link memos.

Typed Files

Most file systems support a small number of built–in file types. The DOMAIN file system, on the other hand, supports an essentially unlimited number of file types; in fact, users can define new file types without changing the underlying file system. Associated with a given file type is a type manager that knows about the internal representation of files of that type. When a program attempts to access a file, the DOMAIN file system automatically invokes the file's type manager to perform the requested operation. See *Using the Open System Toolkit to Extend the Streams Facility* for additional information about type managers.

The NFS protocol does not support the concept of typed files. All files are assumed to be unstructured streams of bytes. This does not generally cause problems for programs that run on foreign systems and want to access DOMAIN files, since their file requests go to the DOMAIN NFS server which then uses the appropriate type manager to access the requested file. Thus, DOMAIN files appear to be unstructured byte streams to a program running on a foreign system.

Many programs running on DOMAIN systems also expect to create and manipulate unstructured files (that is, files of type UASC). These programs can access the unstructured files on foreign systems through the NFS software without difficulty.

A DOMAIN program that attempts to create any file other than a UASC file on a foreign system through the NFS software will not be able to because the NFS protocol does not understand the concept of typed files. For example, the DOMAIN/IX **cp** command and the AEGIS **cpf** and **cpt** commands look at the type of the file being copied and attempt to create a target file of the same type. Since the NFS protocol only allows the commands to create unstructured files, they can only copy UASC files to foreign systems.

You can transfer a non–UASC DOMAIN file to a foreign system by redirecting the output of the AEGIS **catf** command or the DOMAIN/IX **cat** command, both of which read a file as an unstructured stream of bytes and write the byte stream to the target file. However, the resulting file on the foreign system has lost the internal structure of the original DOMAIN file. It simply contains a stream of bytes representing the logical contents of the DOMAIN file. For example, if //dusk_usr is a mount point for the /usr file system of a foreign system and t.bin is a file of type BIN in your DOMAIN working directory, then the DOMAIN/IX command:

```
cat t.bin > //dusk_usr/x.bin
```

creates a file named x.bin in the /usr file system of the foreign system and writes the logical contents of the DOMAIN t.bin file to that file. The t.bin and x.bin files are not identical; the x.bin file does not have the internal structure of the DOMAIN t.bin file. The same would be true for non–UASC DOMAIN files copied to the foreign system using that system's file copy command.

lote that the problems discussed in this section are not bugs but re-
ılt from inherent limitations of the NFS protocol.

:xtended Pathnames

ı the DOMAIN file system, a standard pathname consists of one or
ıore directory names followed by an object name; for example,
bud/tests/test1. An extended pathname in the DOMAIN file sys-
'm is a standard pathname with an extension or residual at the end;
ır example, //bud/tests/test1/12.

xtended pathnames allow you to access all types of objects from
ithin the DOMAIN file system. For example, you can use extended
ıthnames to access the casehm files used by the DOMAIN Software
ngineering Environment (DSEE). A casehm file includes the file's
test version and a history of all of its previous versions. To open
ıd read a casehm file's most recent version, you can use a standard
ıthname such as //bud/tests/test1. To open and read version 12 of
ıch a file, you would need to use the extended pathname
bud/tests/test1/12.

lost NFS implementations on foreign systems can use extended
ıthnames to access all objects within the DOMAIN file system — ex-
:pt casehm files. Foreign systems can use standard pathnames to
:cess the most recent versions of casehm files, but foreign systems
ınnot use extended pathnames such as //bud/tests/test1/12 to access
:evious versions of casehm files. For more information about ex-
nded pathnames, consult *Using the Open System Toolkit to Extend*
:e Streams Facility*.

;uper–User Limitation

he NFS protocol does not allow processes to have super–user privi-
ges across networks. If a super–user process on a foreign system at-
mpts to access a DOMAIN file, it has the same access rights as a
rocess with the subject identifier *user.none.none*. If a super–user
ocess on a DOMAIN workstation attempts to access a file on a for-
gn system, it has the same access rights as the least privileged user
ı that system. On many systems this corresponds to a user ID of *–2*
ıd a group ID of *–2*.

Chapter 3

Accessing Files with Commands

With the DOMAIN NFS software, you can access files in mounted filesystems through the Display Manager and through DOMAIN/IX and AEGIS Shell commands.

Using the Display Manager

The Display Manager (DM) works the same way for files located on foreign hosts as it does for files located on DOMAIN workstations.

Note that you cannot use the DM to create or manage processes on a foreign host. To perform such operations, you must log onto the foreign system with **telnet** or **rlogin** (consult *Using telnet and ftp* and the *DOMAIN/IX Command Reference for BSD4.2* for descriptions of these commands).

Using DOMAIN/IX Commands

The DOMAIN NFS implementation allows you to use almost all of the standard DOMAIN/IX file access and manipulation commands on foreign files. The limitations are as follows:

- DOMAIN/IX commands cannot create, access, or execute DOMAIN object modules located on foreign systems.
- You cannot execute the following commands on files located on foreign systems:

  ld
  nm
  size
  strip

  Also, the output of the **cc** compiler must be sent to a DOMAIN workstation and not to a file located on a foreign system (although the source files can be on a foreign system).
- The **df** command is not supported.

You can use each of the DOMAIN/IX Shell's standard wildcards when accessing objects on a foreign system.

Consult the *DOMAIN/IX Command Reference for BSD4.2* for descriptions of BSD4.2 commands and the *DOMAIN/IX Command Reference for System V* for descriptions of System V commands.

Using AEGIS Shell Commands

The DOMAIN NFS implementation lets you use most of the standard AEGIS Shell commands when working with remote files.

Special Characters in Command Lines

You can execute AEGIS Shell commands that include any of the special characters described in the *DOMAIN System Command Reference*.

The special characters include:

- Pathname wildcards (e.g., ?, %, *, ..., =)
- Input/output control characters (e.g., <, >>, |, ( ))
- Parsing operators (e.g., #, ;, &, ^*, ln, @)

AEGIS Shell Command Support

Except for the commands listed in the following tables, the DOMAIN NFS implementation completely supports all AEGIS Shell commands that access user files.

Table 3-1 contains commands with restricted support; namely, the DOMAIN language compilers. The compilers cannot save their object files on foreign systems.

Table 3-2 contains unsupported commands. These commands have relevance only within the DOMAIN network; for example, the table contains the **crsubs** command which creates a protected subsystem.

Table 3-3 contains commands that currently work only with DOMAIN objects.

In addition, AEGIS commands cannot create, access, or execute DOMAIN object modules located on foreign systems.

Table 3-1. Restricted AEGIS Shell Commands

Command	Description
cc	DOMAIN C compiler
ftn	DOMAIN FORTRAN compiler
pas	DOMAIN Pascal compiler

Table 3-2. Unsupported AEGIS Shell Commands

Command	Description
acl	List or copy an Access Control List
chuvol	Change a volume's UID
cmt	Compare source tree to target tree
crrgy	Create or modify a network registry
crsubs	Create a protected subsystem
ctob	Catalog an object
cvt_rec_uasc	Convert file types
dmtvol	Dismount a logical volume (use umount)
edacct	Create, edit, or list accounts
edacl	Edit/list an Access Control List
edns	Invoke editor for NS_HELPER
edppo	Edit/list person, project, or org. names
ensubs	Enter a protected subsystem
find_orphans	Locate and catalog uncataloged objects
invol	Initialize disk volumes
lkob	Lock an object
llkob	List locked objects
lrgy	List registry sites
mtvol	Mount a logical volume (use mount)
obty	Set or display the type of an object
salacl	Salvage an Access Control List
sald	Salvage a directory
salrgy	Salvage (update) the network registry
salvol	Verify/correct allocation of disk blocks
subs	Set or display subsystem attributes
uctob	Uncatalog pathname, but do not delete
ulkob	Unlock an object
xsubs	Run Shell script subsystem manager

Table 3-3. AEGIS Shell Commands Requiring DOMAIN Objects

Command	Description
bind	Combine obj. modules into executable file
cpboot	Copy the system boot file SYSBOOT
crf	Create a file
crpad	Create a transcript pad and window
debug	Invoke the Language Level Debugger
edfont	Edit a character font
edmtdesc	Edit magtape descriptor file
hpc	Program counter histogram
lbr	Combine object modules into a library
lvolfs	List free space on logical volumes
rbak	Restore/index magnetic media backup file
rwmt	Read/write foreign magtapes
wbak	Create a magnetic media backup file

Chapter 4

Accessing Files from Your Programs

With the DOMAIN NFS implementation, you can use the following programming techniques to access files located in foreign filesystems:

- DOMAIN/IX system calls. (These are identical to the calls supplied by the DOMAIN C libraries.)
- The AEGIS stream and name management system calls. These include ios_$, stream_$, and certain name_$ calls.
- The standard file access and I/O procedures that are part of your programming language; for example, the Pascal READLN or PUT procedures. The DOMAIN NFS software supports the standard I/O procedures of DOMAIN C, FORTRAN, and Pascal.

NOTE: This chapter lists calls and procedures that access and manipulate files and directories. It includes calls that take file

or directory names as input or output arguments. It does not include calls that can only manipulate DOMAIN system files or objects.

DOMAIN C Library Functions and DOMAIN/IX System Calls

The DOMAIN NFS implementation supports both the BSD4.2 and System V DOMAIN C libraries, including the standard library I/O functions and system-level I/O functions.

The DOMAIN NFS software also supports the DOMAIN/IX system I/O calls. These calls are identical to the functions that are included in the DOMAIN C library. Therefore this section covers both the C library and the DOMAIN/IX Calls.

Note that the DOMAIN NFS implementation translates all of the C calls to ios_$ calls and that the C calls default to unregulated access.

Table 4-1 lists standard library I/O (and I/O-related) functions and calls and the system-level library functions and calls. This table indicates whether each call is included in BSD4.2, System V, or both. It also indicates whether each call is fully supported (Y), unsupported (N), or supported with restrictions (R). See the *DOMAIN C (CLIB) Reference* for details about these functions and their use.

Table 4-1. C Function and DOMAIN/IX Call Support

Function	System		Support
	BSD	System V	Y/N/R
Standard Library Functions and Calls			
clearerr	✓	✓	Y
ctermid		✓	Y
fclose	✓	✓	Y
fdopen	✓	✓	Y
feof	✓	✓	Y
ferror	✓	✓	Y
fflush	✓	✓	Y

Table 4-1. C Function and DOMAIN/IX Call Support (Continued)

Function	System		Support
	BSD	System V	Y/N/R
Standard Library Functions and Calls			
fgetc	✓	✓	Y
fgets	✓	✓	Y
fileno	✓	✓	Y
fopen	✓	✓	Y
fprintf	✓	✓	Y
fputc	✓	✓	Y
fputs	✓	✓	Y
fread	✓	✓	Y
freopen	✓	✓	Y
fscanf	✓	✓	Y
fseek	✓	✓	Y
ftell	✓	✓	Y
fwrite	✓	✓	Y
getc	✓	✓	Y
getchar	✓	✓	Y
gets	✓	✓	Y
getw	✓	✓	Y
perror	✓	✓	Y
printf	✓	✓	Y

Table 4-1. C Function and DOMAIN/IX Call Support (Continued)

Function	System BSD	System V	Support Y/N/R
Standard Library Functions and Calls (Continued)			
putc	✓	✓	Y
putchar	✓	✓	Y
puts	✓	✓	Y
putw	✓		Y
rewind	✓	✓	Y
scanf	✓	✓	Y
setbuf	✓	✓	Y
setbuffer	✓		Y
setlinebuf	✓		Y
setvbuf		✓	Y
sprintf	✓	✓	Y
sscanf	✓	✓	Y
vprintf	✓	✓	Y
vsprintf	✓	✓	Y
tempnam		✓	Y
tmpfile		✓	Y
tmpnam		✓	Y
ungetc	✓	✓	Y
vfprintf	✓	✓	Y

Table 4-1. C Function and DOMAIN/IX Call Support (Continued)

Function	System BSD	System V	Support Y/N/R
System Functions and Calls			
access	✓	✓	Y
chdir	✓	✓	Y
chmod	✓	✓	Y
chown	✓	✓	N
close	✓	✓	Y
closedir	✓		Y
creat	✓	✓	Y
default_acl	✓	✓	N
dup	✓	✓	Y
dup2	✓		Y
execl	✓	✓	N
execle	✓	✓	N
execlp	✓	✓	N
execv	✓	✓	N
execve		✓	N
execvp	✓	✓	N
fcntl	✓	✓	Y
flock	✓		N
fstat	✓	✓	Y
fsync	✓		Y
getcwd		✓	Y

Table 4-1. C Function and DOMAIN/IX Call Support (Continued)

Function	System BSD	System V	Support Y/N/R
System Functions and Calls (Continued)			
ioctl	✓		Y
link	✓	✓	Y
lseek	✓	✓	Y
lstat	✓		Y
mkdir	✓		Y
mknod	✓	✓	R
mktemp	✓	✓	Y
open	✓	✓	Y
opendir	✓		Y
read	✓	✓	Y
readdir	✓		Y
readlink	✓		Y
readv	✓		Y
rename	✓		Y
rewinddir	✓		Y
rmdir	✓		Y
seekdir	✓		Y
soft_link	✓	✓	N
soft_unlink	✓	✓	N
stat	✓	✓	Y
symlink	✓		Y

Table 4-1. C Function and DOMAIN/IX Call Support (Concluded)

Function	System BSD	System V	Support Y/N/R
System Functions and Calls (Continued)			
elldir	✓		Y
runcate	✓		Y
umask	✓	✓	Y
unlink	✓	✓	Y
utime	✓	✓	Y
utimes	✓		Y
write	✓	✓	Y
writev	✓		Y

OMAIN C and DOMAIN/IX Call Restriction

he DOMAIN NFS implementation allows the **mknod** function to ake directories but not special files.

AEGIS System Calls

he DOMAIN NFS software supports AEGIS name_$, ios_$, and ream_$ system calls for accessing and modifying files and directoes located in foreign filesystems. (You should use ios_$ calls in new rograms because these calls supercede the stream_$ calls.) However, several functions cannot be fully supported due to the differnces between the AEGIS operating system and other systems. Table -2 lists these calls and indicates whether each call is fully supported Y), unsupported (N), or supported with restrictions (R).

ee the *DOMAIN System Call Reference (Volume II)* for details bout each call. See *Programming with General System Calls* for inormation on how to perform stream I/O.

Table 4-2. AEGIS System Call Support

Call	Support Y/N/R
ios_$ Calls	
ios_$change_path_name	Y
ios_$close	Y
ios_$create	R
ios_$delete	Y
ios_$dup	Y
ios_$equal	Y
ios_$force_write_file	Y
ios_$get	Y
ios_$get_dir	Y
ios_$get_handle	Y
ios_$inq_byte_pos	Y
ios_$inq_conn_flags	Y
ios_$inq_cur_rec_len	Y
ios_$ing_file_attr	Y
ios_$inq_full_key	Y
ios_$inq_mgr_flags	Y
ios_$inq_path_name	Y
ios_$inq_obj_flags	Y
ios_$inq_rec_pos	Y
ios_$inq_rec_remainder	Y
ios_$inq_rec_type	Y
ios_$inq_short_key	Y

Table 4-2. AEGIS System Call Support (Continued)

Call	Support Y/N/R
ios_$ Calls (Continued)	
ios_$inq_type_uid	Y
ios_$locate	Y
ios_$open	Y
ios_$put	Y
ios_$replicate	Y
ios_$seek	Y
ios_$seek_full_key	Y
ios_$seek_short_key	Y
ios_$seek_to_bof	Y
ios_$seek_to_eof	Y
ios_$set_conn_flag	R
ios_$set_dir	Y
ios_$set_locate_buffer_size	Y
ios_$set_obj_flag	R
ios_$set_rec_type	R
ios_$switch	Y
ios_$truncate	Y
name_$ Calls	
name_$add_link	N
name_$cname	N
name_$create_directory	N
name_$create_file	N

Table 4-2. AEGIS System Call Support (Continued)

Call	Support Y/N/R
name_$ Calls (Continued)	
name_$delete_directory	N
name_$delete_file	N
name_$drop_link	N
name_$extract_data	N
name_$get_ndir	Y
name_$get_path	Y
name_$get_wdir	Y
name_$read_dir	N
name_$read_link	N
name_$set_ndir	Y
name_$set_wdir	Y
stream_$ Calls	
stream_$close	Y
stream_$create	Y
stream_$create_bin	N
stream_$create_here	Y
stream_$delete	Y
stream_$force_write_file	Y
stream_$get_buf	Y
stream_$get_conditional	Y
stream_$get_ec	Y
stream_$get_prior_rec	Y
stream_$get_rec	Y

Table 4-2. AEGIS System Call Support (Concluded)

Call	Support Y/N/R
stream_$ Calls (Continued)	
stream_$inquire	Y
stream_$open	Y
stream_$put_chr	Y
stream_$put_rec	Y
stream_$redefine	R
stream_$replace	Y
stream_$seek	Y
stream_$truncate	Y

AEGIS Call Restrictions

The following sections describe the restrictions on each AEGIS call that has limitations when applied to files located in foreign filesystems.

ios_$create Restrictions

With ios_$create, you may specify any valid object type, but the DOMAIN NFS software creates the type according to Table 4-3.

Table 4-3. DOMAIN NFS Object Types and ios_$create

Type Requested	Type Created	Description
directory_$uid	nfs_dir_$uid	A remote directory
uasc_$uid	nfs_file_$uid	A remote, unstructured file
Any other type	None. Returns the ios_$ill_obj_type error.	Error

ios_$set_conn_flag Restrictions

With ios_$set_conn_flag, only file access is allowed. Therefore, the following connection flags cannot be set to on; any attempt to set them will result in an ios_$illegal_operation error:

tty
ipc
vt

ios_$set_obj_flag Restrictions

With ios_$set_obj_flag, an attempt to set an object flag of ios_$of_sparse_ok to false is ignored.

ios_$set_rec_type Restrictions

With ios_$set_rec_type, an attempt to set the record type to anything other than ios_$undef will result in an ios_$illegal_operation error.

stream_$redefine Restrictions

The following restrictions apply to the attributes that you can set by using the stream_$redefine call for a directory or file located in a foreign filesystem.

Object Types
: You cannot redefine the object type.

Sparse Flag
: You cannot set the Sparse Flag false.

Standard I/O Procedures

This section describes the Pascal I/O procedures and FORTRAN I/O statements supported by the DOMAIN NFS software.

DOMAIN Pascal I/O Procedures

The DOMAIN NFS implementation fully supports the following standard predeclared DOMAIN Pascal input/output procedures. See the *DOMAIN Pascal Language Reference* for details about these procedures and their use.

- close
- find
- get
- open
- page
- put
- read
- readln
- replace
- reset
- rewrite
- write
- writeln

DOMAIN FORTRAN I/O Statements

In addition, the DOMAIN NFS implementation fully supports the following standard DOMAIN FORTRAN input/output statements. See the *DOMAIN FORTRAN Language Reference* for details about these procedures and their use.

- backspace
- close
- endfile
- inquire
- open
- print

- read
- rewind
- write

Programmer's Notes

Because of the way foreign systems manipulate files, and because of the DOMAIN NFS implementation itself, there are several considerations that affect program performance. The following notes list some information that you should keep in mind when writing programs that access files located in foreign filesystems.

- Byte seeks are faster than Record seeks.
- The ios_$inq_rec_pos call requires significant internal processing to determine the *end of file* location.
- The DOMAIN NFS implementation does not support many of the name_$ calls (see Table 4–2). Use the corresponding ios_$ calls in their place, as indicated in Table 4–4.

Table 4–4. ios_$ Equivalents for name_$ Calls

name_$ Call	ios_$ Call
name_$create_file	ios_$create
name_$create_directory	ios_$create
name_$delete_file	ios_$delete
name_$delete_directory	ios_$delete
name_$read_dir	ios_$get

- When you use ios_$get to read a directory located in a foreign filesystem, the call returns a 44–character DOMAIN directory record.
- You cannot write to a directory (nfs_dir) or gateway object (nfs_gate). However, you can create new files in the directory by using the ios_$create call.

Appendix A

Descriptions of Commands, Servers, and Files

This appendix contains descriptions of the following DOMAIN NFS commands, server daemons, and reference files, arranged in alphabetical order (you can also access these descriptions from BSD4.2 and System V Shells with the **man** command):

exports
fstab
mount/umount
mountd
mtab
nfsd
nfsstat
portmap
rpcinfo
showmount

exports

Name

exports — specify mountable file systems

Usage

/etc/exports

Description

The /etc/exports file contains a vertical list of left–justified file system names. On each line, following the file system name and separated by spaces, are the name(s) of the host(s) authorized to remote mount that file system. If a line contains a file system name and no host names, then any host can remote mount the specified file system.

When **mountd(8c)** processes a mount request, it checks /etc/exports for the export status of the file system and /etc/hosts to validate the hosts associated with that file system. A system administrator normally maintains /etc/exports and its ACL (Access Control List).

A pound sign (#) in the file identifies the characters to the right of the pound sign and to the end of the line as a comment.

Examples

```
//jodhpurs                          #everyone
//jodhpurs        roscoe            #just roscoe
//jodhpurs        cameo jubilant primo    #just these three
```


Files

/etc/exports

Related Information

mountd(8c)

fstab

Name

fstab — table of remote file systems used by local host

Usage

read by mount and umount commands

Description

The /etc/fstab file contains information about file systems mounted by the local host. Users maintain this file by producing formatted file system entries with the mount command and its -p option, then manually adding the entries to the /etc/fstab file with a text editor.

The **mount** and **umount** commands process the entries from top to bottom. The entries must therefore appear in proper sequence; i.e., a file system mounted within another file system must appear below that file system.

An entry occupies a single line and contains the following information:

host:fsname gwname type args **0 0**

where:

host:fsname	remote host name, colon (:), and file system name
gwname	gateway object pathname (remote file systems) or temporary directory pathname (removable file systems)

type	file system type specified at mount time with mount command's -t option (possible values are "4.2" and "nfs"; default is "nfs")
args	argument(s) specified at mount time with mount command's -o option (default is "hard")
0 0	Unused

Entries in the /etc/mtab file have the same format.

Example

bonmot:/usr /bonmot_usr nfs soft 0 0

Files

/etc/fstab

Related Information

mount(8)
mtab(5)

mount umount

Name

mount, umount — mount and dismount file system

Usage

removable file systems only

/etc/mount [–ar] [*dev dir*]

/etc/umount [–a] [*dev*]

remote file systems only

/etc/mount [–afrvp] [to *type options*] [*fsys*] [*dir*]

/etc/umount [–av] [–root] [*fsys* | *dir*]

Description

The **mount** command mounts removable and remote file systems.

The **umount** command unmounts removable and remote file systems.

When mounting a *removable* file system located on a diskette, cartridge tape, or magtape, specify the device name for *dev* and a nonexistent directory pathname for *dir*.

When mounting a *remote* file system (that is, one located on another host), specify the remote host and file system in the form *host:fsname* for *fsys* and a nonexistent directory for *dir*. (The

mount command creates a gateway object named *dir*; the umount command deletes it.)

The **mount** and **umount** commands maintain a table of mounted file systems in /etc/mtab. If you execute the mount command without arguments, it prints the table. If you specify only one argument, the command searches /etc/fstab for an entry with the same argument; if the entry exists, the command then uses it to mount the file system.

Options

The command options can be one or more of the following items separated by a space:

–a With **/etc/mount**, try to mount all file systems listed in the file /etc/fstab. (This file contains entries added manually by user.)

With **/etc/umount**, try to dismount all file systems listed in /etc/mtab (this file contains an entry for every remote file system currently mounted by the local host).

–f Fake a new /etc/mtab entry without mounting the file system.

–o Accept the following arguments separated by commas, as options. The defaults are fg, retry=1, timeo=7, and hard:

bg	Retry in background if server host's **mountd**(8c) doesn't respond.
fg	Retry in foreground.
hard	Retransmit until server responds.

port=*n* Set NFS port number *n*.

retry=*n* Set number of mount retries to *n*.

ro Allow read only.

root Mount in global root directory.

rsize=*n* Set read request buffer size to *n* bytes.

rw Allow read/write.

soft Report error if server fails to respond.

timeo=*n* Set NFS timeout to *n* tenths of a second. When attempting to access mounted file system, NFS waits specified amount of time for response; if no response, NFS multiplies *n* by 2 and retransmits request.

wsize=*n* Set write request buffer size to *n* bytes.

-p List the mounted file systems in the same format as entries in /etc/fstab.

-r Mount the file system as read-only.

-root Dismount the file system in the global root directory.

-t Accept the following argument as the file system type; the recognized types are "4.2" and "nfs".

-v Display a message when mounting the file system.

Files

/etc/mtab

/etc/fstab

Cautions

Note that physically write-protected file systems, as well as those on magnetic tape, must be mounted read-only. If not, errors develop even if no explicit write is attempted.

Mounting a removable file system that contains unreadable or otherwise invalid data can have unpredictable consequences.

Related Information

mount(2)
mtab(5)
fstab(5)
mkdisk(8)

mountd

Name

mountd — manages file system mount requests

Usage

cps /etc/mountd [-*sn*]

Description

The **mountd** server daemon runs on a server host and manages file system mount requests. When it receives a request from a client host, it consults the server host's /etc/exports(5) file to determine if the client host can access the requested file system. **Mountd** also supplies the **showmount(8)** command with information about client hosts that have file systems mounted.

Normally, you invoke the server daemon in the server host's `node_data/startup file. The -s option puts the server daemon to sleep for *n* seconds to allow the TCP server to fully initialize before the server daemon requires its resources. The recommended value for *n* is 15.

Example

cps /etc/mountd -s15

Related Information

nfsd(8c)
portmap(8c)

mtab

Name

mtab — table of mounted file systems

Usage

used by mount and umount commands

Description

The /etc/mtab file identifies the file systems currently mounted by the local host. The **mount** command creates entries in this file; the **umount** command deletes them.

Each entry corresponds to a file system currently mounted by the local host. An entry occupies a single line and contains the following information:

host:fsname gwname type args **0 0**

where:

host:fsname	remote host name, colon (:), and file system name
gwname	gateway object pathname (remote file systems) or temporary directory pathname (removable file systems)
type	file system type specified at mount time with mount command's -t option (possible values are "4.2" and "nfs"; default is "nfs")
args	argument(s) specified at mount time with mount command's -o option (default is "hard")

0 0 Unused

Example

bonmot:/usr /bonmot_usr nfs soft 0 0

Files

/etc/mtab

Related Information

mount(8)
fstab(8)

nfsd

Name

nfsd — services file system mount requests

Usage

cps /etc/nfsd [-s*n*]

Description

The **nfsd** server daemon runs on a server host and services the file system mount requests of client hosts.

Normally, you invoke the server daemon in the server host's `node_data/startup file. The -s option puts the server daemon to sleep for *n* seconds to allow the TCP server to fully initialize before the server daemon requires its resources. The recommended value for *n* is 15.

Example

cps /etc/nfsd -s15

Related Information

mountd(8c)
portmap(8c)

nfsstat

Name

nfsstat — Network File System statistics

Usage

/etc/nfsstat [–csnrz]

Description

The **nfsstat** command displays statistical data about NFS; you can also use it to reinitialize the statistics. Without options, the command defaults to

nfsstat –csnr

Options

–c Display client statistics. You can combine this option with the –n and –r options to print only client NFS or client RPC statistics.

–s Display server statistics in same manner as –c option.

–n Display NFS statistics for both client and server. You can combine this option with the –c and –s options to print only client or server NFS statistics.

–r Display RPC statistics in same manner as –n option.

–z Zero (reinitialize) statistics. You can combine this option with any of the other options to reinitialize particular sets of statistics after printing them. This option does not require super–user privileges.

portmap

Name

portmap — maps RPC program numbers to DARPA port numbers

Usage

cps /etc/portmap [-s*n*]

Description

The **portmap** server daemon runs on a server host and directs data transmissions from client hosts to RPC (Remote Procedure Call) servers. When an RPC server begins operation, it tells **portmap** the port number being monitored and the program numbers being serviced by the RPC server. A client host sending RPC packets to a particular program number gets the appropriate port number from **portmap**.

Normally, you invoke the server daemon in the server host's `node_data/startup file. The -s option puts the server daemon to sleep for *n* seconds to allow the TCP server to fully initialize before the server daemon requires its resources. The recommended value for *n* is 10.

Example

cps /etc/portmap -s10

Related Information

mountd(8c)
nfsd(8c)

rpcinfo

Name

rpcinfo — report RPC information

Usage

/etc/rpcinfo -p [*host*]

/etc/rpcinfo -u *host program_number version_number*

/etc/rpcinfo -t *host program_number version_number*

Description

The **rpcinfo** command makes an RPC call to an RPC server and reports the results.

Options

-p Probe the portmapper on *host*, and print a list of all registered RPC programs.

-u Make an RPC call to procedure 0 of *program number* using UDP, and report if response received.

-t Make an RPC call to procedure 0 of *program number* using TCP, and report if response received.

showmount

Name

showmount — show all remote mounts

Usage

/etc/showmount [**-a**] [**-d**] [**-e**] [*host*]

Description

The **showmount** command lists all remote clients that have mounted file systems belonging to *host*. The command gets its information from the **mountd**(8c) server on host.

Options

-d List directories that have been remotely mounted by clients.

-a Print all remote mounts in the format

hostname:directory

where *hostname* is the name of the client, and *directory* is the root of the mounted file system.

-e Print the list of exported file systems in /etc/exports.

Files

/etc/mtab	mount table
/etc/fstab	file system table
/etc/rmtab	remote mount table

Cautions

If a client crashes, its mount points remain in effect until someone reboots the client and executes the **/etc/umount** command.

Related Information

mountd(8)
exports(5)

Appendix B

Managing NFS on a DOMAIN Network

Before installing the DOMAIN NFS software, you need to

- Determine the optimal server arrangement for your site
- Establish uniform user and group IDs for all NFS hosts

Arranging Servers on the DOMAIN Network

With most NFS implementations, a user intending to access multiple remote file systems must individually mount each file system. In ad-

dition, the hosts on which the remote file systems reside must be server hosts. In Figure B–1, users at hosts fs1, fs2, and fs3 can access each other's file systems because all of their hosts are server hosts.

With DOMAIN NFS software, you could similarly make every workstation on the DOMAIN network a server host. Doing so would make all remote file systems on the network available to users at hosts fs1, fs2, and fs3 — and vice versa. Figure B–1 illustrates this arrangement.

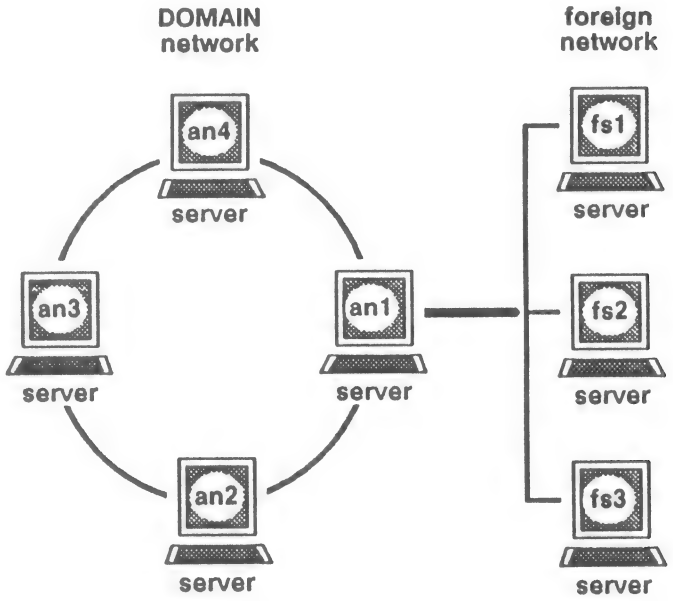

Figure B–1. Sample Server Arrangement

The user at host `fs1` could then use the following mount commands to mount the remote / file systems in both networks. Note that the user would probably need many more mount commands to mount all file systems in the foreign network.

```
/etc/mount -o soft fs2:/ /fs2_root
/etc/mount -o soft fs3:/ /fs3_root
```



```
/etc/mount -o soft an1:/ /an1_root
/etc/mount -o soft an2:/ /an2_root
/etc/mount -o soft an3:/ /an3_root
/etc/mount -o soft an4:/ /an4_root
```

The users at hosts fs2 and fs3 would also need to mount the remote slash (/) file systems in both networks. Note that all three users would have needed to create and/or set aside one directory for every mount command executed (a requirement of other NFS implementations).

The scenario we have described thus far is what the users at hosts fs1, fs2, and fs3 *could* do with DOMAIN NFS software installed throughout the DOMAIN network. However, what they *need* to do in this situation is actually much easier, as you will see in the next section.

Single DOMAIN NFS Server

The DOMAIN network root directory (//) allows users at workstations anywhere on the DOMAIN network to access directories and files anywhere else in the network — without the users having to explicitly mount file systems to do so. DOMAIN NFS software offers a similar capability to users at foreign hosts with the single server configuration.

Figure B–2 illustrates the single DOMAIN NFS server configuration. With this configuration, a user at a foreign host can access remote file systems anywhere on the DOMAIN network by mounting the network root directory through the DOMAIN server host.

For example, the user at host fs1 can access the slash (/) file systems of all hosts on the networks in Figure B–2 after issuing the following mount commands:

```
/etc/mount -o soft fs2:/ /fs2_root
/etc/mount -o soft fs3:/ /fs3_root
/etc/mount -o soft an2:// /apollo
```

As with the previous section, the first two commands explicitly mount the slash (/) file systems located on the foreign hosts fs2 and fs3. However, the third command now provides access to all hosts on the DOMAIN network. In particular, the command mounts the network root directory (//) as /apollo through the server host an2, thereby enabling the user at host fs1 to access an object anywhere on the DOMAIN network by using /apollo as a prefix to the object's pathname. For example, the user can now access the file //an3/test with

the pathname /apollo/an3/test. This approach offers a significant reduction in the time, effort, and the number of directories needed by other NFS ilmplementations to mount a large number of DOMAIN file systems.

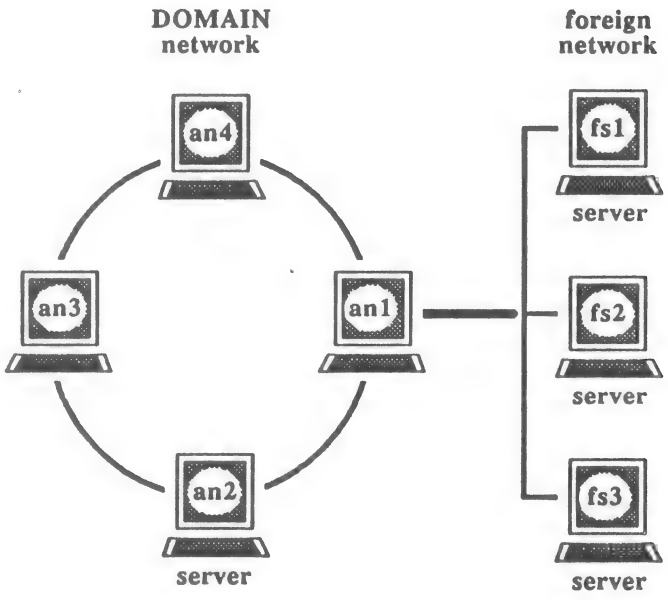

Figure B–2. Single DOMAIN NFS Server Configuration

The network root directory also simplifies the mount procedure for DOMAIN NFS users wanting to mount remote file systems located on foreign hosts. Once a DOMAIN NFS user mounts a remote file system in the network root directory, all DOMAIN NFS users can access that file system by using the name specified in the mount command — without having to issue their own mount commands.

For example, suppose the user at host an1 issues the following mount command:

```
/etc/mount -o soft,root fs2:/usr //fs2_usr
```


Thereafter, any DOMAIN NFS user who wants to access the /usr file system on fs2 only needs to use the pathname //fs2_usr. Note that the DOMAIN NFS mount command must include the –o root option to mount a file system in the root directory.

Multiple DOMAIN NFS Servers

A single DOMAIN NFS server host is sufficient for many sites. However, your site may require multiple DOMAIN NFS server hosts to efficiently handle the needs of users at foreign hosts. The optimum ratio of users to server hosts varies with each site.

When you have multiple server hosts on a DOMAIN network, you should divide all users requiring remote access to DOMAIN file systems into groups — with one group for each server host. All users in the same group should then mount the network root directory through the same server host. The result: groups of users accessing DOMAIN file systems from different points on the network.

For example, Figure B–3 illustrates a DOMAIN network with multiple server hosts. If the users at foreign hosts fs1 and fs2 mount the network root directory through the DOMAIN NFS server host an4 with the command:

```
/etc/mount -o soft an4:// /apollo
```

and the user at foreign host fs3 uses host an2:

```
/etc/mount -o soft an2:// /apollo
```

then all three users can access file systems anywhere on the DOMAIN network by using pathnames beginning with /apollo — even though the users mounted the network root directory at different locations on the network.

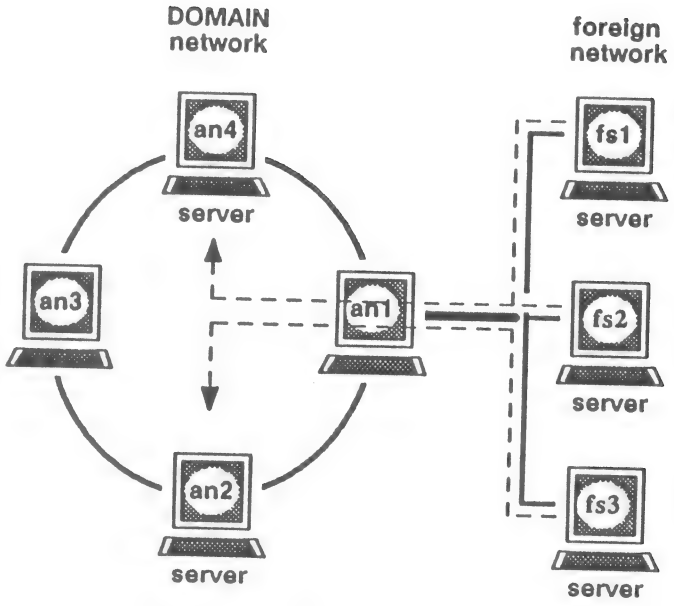

Figure B–3. Multiple DOMAIN NFS Server Hosts

Creating Uniform User and Group IDs

In addition to arranging NFS servers on the DOMAIN network, you also need to ensure that the NFS users at your site have uniform user and group IDs on all of the NFS hosts.

This section describes the three categories of sites requiring DOMAIN NFS software; it also describes the procedures necessary for establishing uniform user and group IDs at those sites.

The following assumptions underly the discussion in this section:

- All potential NFS hosts on an established DOMAIN network already run DOMAIN/IX software (a prerequisite for installing NFS software) and thus have uniform user and group IDs.

- All NFS hosts on an established foreign network also have uniform user and group IDs.

- A site's users have accounts within established networks and not new networks.

In addition, all sites requiring DOMAIN NFS software belong to one of the following categories (see Figure B–4):

A. Sites connecting a new foreign network to an established DOMAIN network

B. Sites connecting a new DOMAIN network to an established foreign network

C. Sites connecting an established DOMAIN network to an established foreign network

To establish uniform user and group IDs at your site, you need to update the /etc/passwd and /etc/group files used by the DOMAIN and foreign networks. Updating these files involves making copies of the files, editing the copies, and then executing the DOMAIN/IX rpasswd command and/or equivalent Unix commands.

The whole process is quite simple. Just determine the category of your site (A, B, or C) and then execute the procedure that corresponds to the category. Descriptions of the procedures begin with the following section.

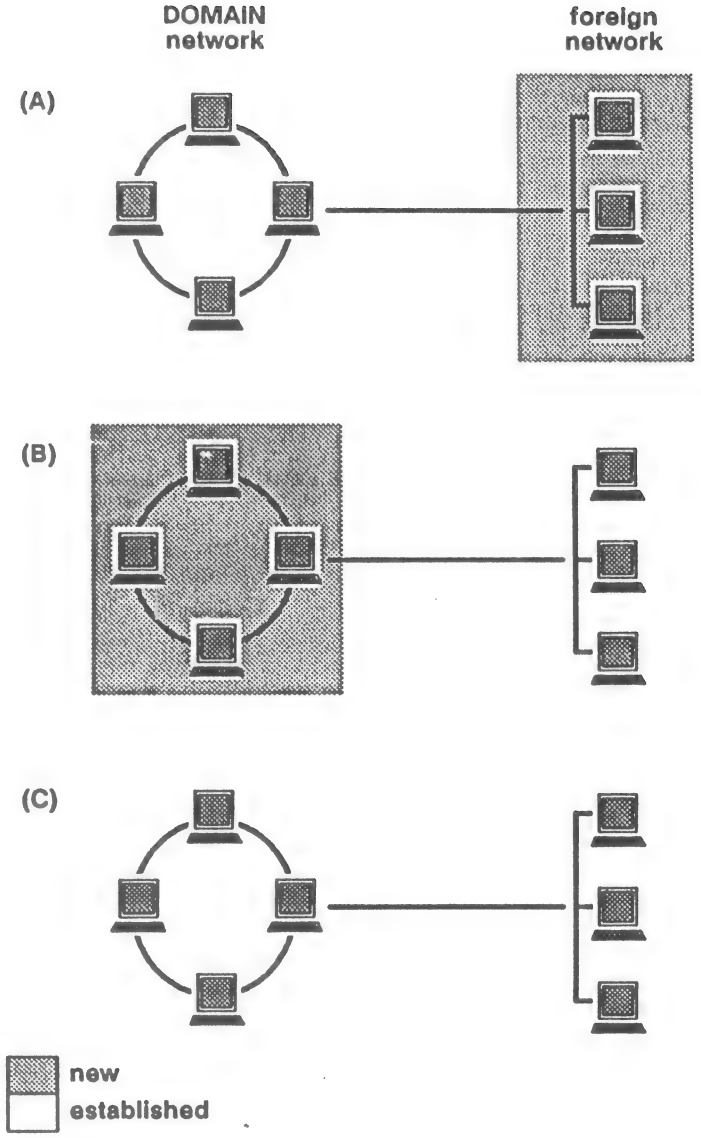

Figure B–4. Categories of Sites Requiring DOMAIN NFS Software

Procedure for Category A Sites

The procedure described in this section assumes that you have just connected a new foreign network to an established DOMAIN network, and that the systems in the foreign network are running NFS software. The procedure also assumes that users at your site do not have accounts on any system in the foreign network; if this is not the case, then use the procedure for Category C sites. Figure B–5 shows a sample Category A site.

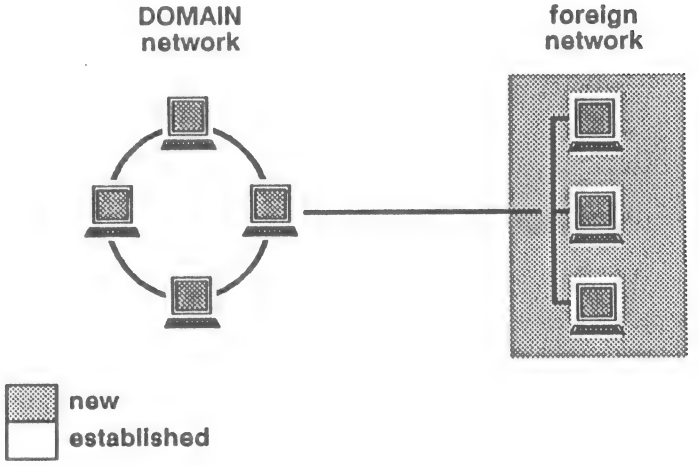

Figure B–5. Category A Site

PROCEDURE B–1. Establishing Uniform User and Group IDs at a Category A Site

❑ Make a copy of the DOMAIN/IX /etc/group file and delete the fifth field from each line of the copy. A line consists of five fields; the fifth field begins with the fourth colon (::::) and includes all text to the end of the line. For example, the line:

bmkt::46::1F367812.600001A2, BASE MARKETING

should read like the following after you delete the fifth field:

```
bmkt::46:
```

- ❑ Copy the /etc/passwd and the modified /etc/group files from the DOMAIN network to each of the systems on the foreign network. Replace the original files on the foreign systems with the copies.

  To copy the files, use the **ftp** (File Transfer Program) command documented in *Using telnet and ftp* and in the *DOMAIN/IX Command Reference for BSD4.2*. The command requires you to enter a user name and password for the foreign system receiving the files. You must therefore have access to a valid account on each foreign system before executing the command.

- ❑ Ensure that every user listed in the /etc/passwd file has a home directory on each foreign system. Create home directories as necessary.

- ❑ Edit the /etc/passwd files on the foreign systems to ensure that the files list the correct home directories for the users.

- ❑ All DOMAIN users now have accounts on the foreign systems but no passwords. Instruct the users to use the **passwd** command to establish passwords when they first log in to the foreign systems.

END OF PROCEDURE B-1.

Procedure for Category B Sites

The procedure described in this section assumes that you have just connected a new DOMAIN network to an established foreign network. The procedure also assumes that users at your site do not have accounts on any workstation in the DOMAIN network; if this is not the case, then use the procedure for Category C sites. Figure B-6 shows a sample Category B site.

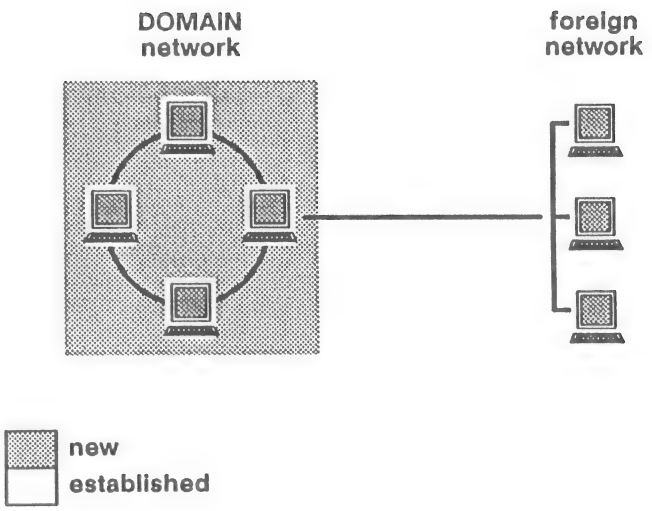

Figure B-6. Category B Site

PROCEDURE B-2. Establishing Uniform User and Group IDs at a Category B Site

- ❑ Add all users and groups on the foreign network to the DOMAIN network registry, with each group becoming a DOMAIN project. Refer to *Administering Your DOMAIN System* for directions.

- ❑ Copy the /etc/passwd and /etc/group files from a system in the foreign network to the /etc directory of the primary DOMAIN/IX system administration node on the DOMAIN network. Replace the original files in the DOMAIN/IX /etc directory with the copies.

  To copy the files, use the **ftp** (File Transfer Program) command documented in *Using telnet and ftp* and in the *DOMAIN/IX Command Reference for BSD4.2*. The command requires you to enter a user name and password for the foreign system contributing the files. You must therefore have access to a valid account on the foreign system before executing the **ftp** command.

- ❑ Run the DOMAIN/IX /etc/crpasswd command documented in *System Administration for DOMAIN/IX BSD4.2* and *System Administration for DOMAIN/IX System V.*
- ❑ Distribute the new /etc/passwd, /etc/passwd.map, and /etc/group files to the slave DOMAIN/IX system administration nodes in the DOMAIN network (see *System Administration for DOMAIN/IX BSD4.2 or System Administration for DOMAIN/IX System V).*

END OF PROCEDURE B-2.

Procedure for Category C Sites

The procedure described in this section assumes that you have just connected an established DOMAIN network to an established foreign network. Figure B-7 shows a sample Category C site.

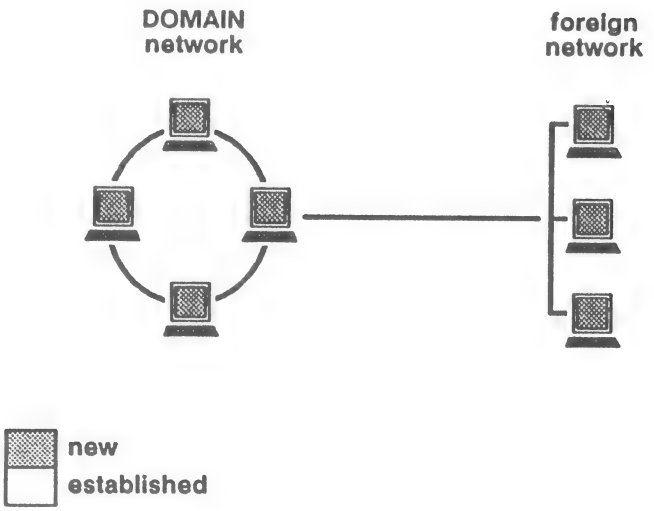

Figure B-7. Category C Site

PROCEDURE B-3. Establishing Uniform User and Group IDs at a Category C Site

- ❑ Compare the /etc/group files for the DOMAIN and foreign networks.
- ❑ If a group exists on the foreign network and not on the DOMAIN network, then add the group to the DOMAIN network registry as a DOMAIN project. Refer to *Administering Your DOMAIN System* for directions.

  If a group exists on the DOMAIN network and not on the foreign network, then add the group to all of the /etc/group files on the foreign network. The group ID that you specify for the new group must differ from all group IDs already used in the foreign network.

  Don't be concerned if the /etc/group files in the foreign and DOMAIN networks have different group IDs for the same group or the same group ID for different groups. A later step in this procedure resolves such discrepancies.
- ❑ Compare the /etc/passwd files for the DOMAIN and foreign networks.
- ❑ If a user has an account in the foreign network and not in the DOMAIN network, then add the user to the DOMAIN network registry. Refer to *Administering Your DOMAIN System* for directions.

  If a user has an account in the DOMAIN network and not in the foreign network, then add the user to the /etc/passwd files in the foreign network. The user ID that you specify for the new user must differ from all user IDs already used in the foreign network. Also assign the user a group ID from the modified /etc/group file on the foreign network.

  Don't be concerned if the /etc/passwd files in the foreign and DOMAIN networks have different user IDs for the same user or the same user ID for different users. A later step in this procedure resolves such discrepancies.
- ❑ Copy the /etc/passwd and /etc/group files from a system in the foreign network to the `/etc` directory of the primary DOMAIN/IX system administration node on the DOMAIN network. Replace the original files in the DOMAIN/IX /etc directory with the copies.

To copy the files, use the **ftp** (File Transfer Program) command documented in *Using telnet and ftp* and in the *DOMAIN/IX Command Reference for BSD4.2*. The command requires you to enter a user name and password for the foreign system contributing the files. You must therefore have access to a valid account on the foreign system before executing the **ftp** command.

- ❑ Update the /etc/passwd and /etc/group files on the DOMAIN network with the DOMAIN/IX **/etc/crpasswd** command documented in *System Administration for DOMAIN/IX BSD4.2* and *System Administration for DOMAIN/IX System V*. This step makes user and group IDs uniform across the foreign and DOMAIN networks.
- ❑ Distribute the new /etc/passwd, /etc/passwd.map, and /etc/group files to the slave DOMAIN/IX system administration nodes in the DOMAIN network (see *System Administration for DOMAIN/IX BSD4.2)*.
- ❑ Execute the DOMAIN/IX **flush_cache** command on all workstation in the DOMAIN network; consult *System Administration for DOMAIN/IX BSD4.2* for a description of the command. Shut down and reboot each workstation after executing the command.

END OF PROCEDURE B-3.

Adding New Users and Groups

Once you've established uniform user and group IDs for the DOMAIN and foreign networks at your site, you can use the following procedures to add new users and groups.

PROCEDURE B-4. Adding a New User to the DOMAIN and Foreign Networks

- ❑ Add the new user to the DOMAIN network registry. Refer to *Administering Your DOMAIN System* for directions.
- ❑ Update the DOMAIN/IX /etc/passwd file on the DOMAIN network with the DOMAIN/IX **/etc/crpasswd** command documented in *System Administration for DOMAIN/IX BSD4.2* and *System Administration for DOMAIN/IX System V.*

- ❑ Add the new user to the /etc/passwd files in the foreign network. Assign the user the same user and group IDs that appear in the new DOMAIN /etc/passwd file.

END OF PROCEDURE B-4.

PROCEDURE B-5. Adding a New Group to the DOMAIN and Foreign Networks

- ❑ Add the new group to the DOMAIN network registry. Refer to *Administering Your DOMAIN System*, *System Administration for DOMAIN/IX BSD4.2*, or *System Administration for DOMAIN/IX System V* for directions.
- ❑ Update the DOMAIN/IX /etc/group file on the DOMAIN network with the DOMAIN/IX **crpasswd** command documented in *System Administration for DOMAIN/IX BSD4.2* and *System Administration for DOMAIN/IX System V.*
- ❑ Add the new group to the /etc/group files in the foreign network. Assign the same group ID that appears in the new DOMAIN /etc/group file.

END OF PROCEDURE B-5.

Appendix C

Troubleshooting Guide

This appendix describes problems that may result from improper installation or use of the NFS software and provides solutions whenever possible. Table C–1 describes problems that you may encounter while at a DOMAIN host; Table C–2 describes problems that you may encounter while at a foreign host.

Table C-1. Possible Problems at a DOMAIN Host

Cannot edit files with invisible names (e.g., ".login")	You're unable to use the Display Manager to edit a remote file that has an invisible filename (one beginning with a "."). Enclose the file's pathname in single quotation marks (' ').
Cannot mount a remote file system	You're unable to mount a remote file system. Check the /etc/exports file on the host where the file system resides. If the file does not give you access to the file system, then either edit the file or ask someone to do it for you. Another possibility is that the foreign host where the file system resides isn't running the NFS server. If so, start the following server daemons on the host in the specified order: **/etc/portmap**, **/etc/mountd**, and **/etc/nfsd**.
Cannot mount a remote file system from a gateway node	When you attempt to mount a remote filesystem from a gateway node, you get an error message that reads '"mount: not in export list for %s:%s\n",rhost, rdir).' Check the /etc/exports file on the host where the file system resides. If the file gives the gateway node access to the file system, then check the /etc/hosts file on the DOMAIN network. The file should list the gateway node's address for the foreign network first and the gateway node's DOMAIN address second.
Cannot start the NFS server from 'node_data/startup	A DOMAIN server host has the correct NFS software and a `node_data/startup file that invokes the NFS server daemons **/etc/portmap**, **/etc/mountd**, and **/etc/nfsd**. However, the NFS server daemons won't run properly when you reboot the host. Edit the host's `node_data/startup file and replace the lines invoking the NFS server daemons with the following lines: cps /etc/portmap –s10 cps /etc/mountd –s15 cps /etc/nfsd –s15 The *–sn* option puts the NFS server daemons to sleep for *n* seconds, thereby allowing the TCP server to fully initialize before the daemons require its resources.
Directory appears to be empty	You attempted to access a remote directory through a mounted file system and the directory appears to be empty, though you know it's not. The directory may actually be a different file system. Try mounting it with the **/etc/mount** command.

Table C-2. Possible Problems at a Foreign Host

Cannot access DOMAIN objects, but others can	You cannot access objects on the DOMAIN network from a foreign system, even though others can access the same DOMAIN objects from the same foreign system. Compare the /etc/passwd files on the foreign system and the DOMAIN network. If you have different user and group IDs in the two files, ask your system administrator to resolve the differences.
Cannot access some DSEE objects	You cannot access DSEE casehm files on the DOMAIN network with extended pathnames like //node/dir/file/version_number. The NFS software does not access such objects; see the "Type Files" section in Chapter 2 for details. To access such objects

from a foreign system, you need to log in to a DOMAIN workstation with the **telnet** or **rlogin** command.

Cannot mount a remote file system

You're unable to mount a remote file system.

Check the /etc/exports file on the host where the file system resides. If the file doesn't give you access to the file system, then either edit the file or ask someone to do it for you.

Another possibility is that the DOMAIN host used to mount the DOMAIN file system isn't running the NFS server. If so, start the following server daemons on the host in the specified order: **/etc/portmap**, **/etc/mountd**, and **/etc/nfsd**.

DOMAIN file briefly stays locked after NFS access

The NFS server waits 3 to 6 seconds after accessing a DOMAIN file to unlock it. This allows the NFS server to reduce the number of times it opens and closes the same file.

ls -l of // stalls the NFS server

You've used the **ls -l** command to list the contents of the DOMAIN network root directory (//) through a DOMAIN server host. The host's NFS server consequently stalls.

The **ls** command encountered an inaccessible object in the network root directory, causing the NFS server to stall for a brief period of time; as a result, the NFS server stops processing other requests during that time. Since at any given time an object in the DOMAIN network root directory may be temporarily inaccessible, you should avoid using the **ls -l** command on that directory.

NFS server active but unresponsive

You've tried to access a remote file system on a DOMAIN host but its NFS server does not respond. When you execute the **pst** AEGIS Shell command, it indicates that the DOMAIN host's NFS server is active.

Examine the `node_data/nfs_error_log file on the DOMAIN host. You should find an entry that reads "NFSD: main()..FYI..Fault received. Object not found in routine 'UNIX_UID_$UNIX_ID_TO_UID' line 334." If so, restart the NFS server.

Appendix D

Error Messages

This appendix lists the error messages issued by the **/etc/mount** and **/etc/umount** commands in alphabetical order.

Note that the *italicized text* simply indicates the type of information that appears in an error message. You should therefore consider only the **boldfaced text** when looking for a message in this appendix.

mount: *pathname* already exists
mount: *pathname* already exists in root

You tried to create the mount point *pathname* in the network root directory with the **/etc/mount** command and the –o root option; however, *pathname* already exists. The command requires a nonexistent pathname.

mount: directory path '*pathname*' must begin with '/'

You invoked the **/etc/mount** command and specified the invalid *pathname* as the mount point. The pathname of a mount point must begin with a slash (/) as in //chauncy/mntpt or /mntpt.

mount: gateway object *pathname* already exists

The **/etc/mount** command could not create the gateway object or mount point *pathname* because it already exists. If the gateway object or mount point is a directory, the following message also appears.

Domain mounts cannot be performed on an existing object. The gateway object is a file created by the mount command.

DOMAIN NFS software creates a gateway object (i.e., a file) when you mount a remote file system; most other NFS implementations use an existing directory.

mount: *cmd_input* is an improper NFS FileSystem name

You invoked the **/etc/mount** command and specified the unknown file system *cmd_input*. The NFS protocol requires file system names in the format 'host:filesystem.'

mount: '*options*' is an invalid option list

You invoked the **/etc/mount** command with the -o option and specified the unrecognized *options*.

mount: *pathname* is not a root

You invoked the **/etc/mount** command with the -o root option and specified the invalid *pathname* as the mount point. A mount point specified with the -o root option must begin with double slashes (//).

mount: *host:fs* on *pathname*: No such file or directory

You attempted to mount the remote file system *host:fs* at the mount point *pathname* with the **/etc/mount** command. The file system *fs* does not exist on the remote host *host*.

mount: not in export list for *host:fs*

Your local host (i.e., the system that printed this message) does not have access rights to the file system *host:fs*. The /etc/exports file on the server host *host* determines which client hosts can access which of its file systems.

mount: rpc error \"*message*\"

The RPC (Remote Procedure Call) error described by *message* occurred during your attempt to mount a remote file system with the **/etc/mount** command.

mount: *host:fs* server not responding: RPC_TIMED_OUT

You attempted to mount the remote file system *host:fs* with the **/etc/mount** command. However, the NFS server responsible for that file system didn't respond so the command aborted.

mount: unable to create *pathname* in local root

The **/etc/mount** command could not create the mount point entry *pathname* in the local copy of the network root directory. Check the ACL of /registry/local_registry and try the command again.

mount: unknown host *cmd_input*

You invoked the **/etc/mount** command and specified the unknown host *cmd_input*. The file /etc/hosts lists all valid hosts.

pathname not mounted

You invoked the **/etc/umount** command and specified a file system or mount point not found in the host's /etc/mtab file.

umount: must specify -root for unmounting *pathname*

You didn't use the -root option with the **/etc/umount** command to dismount a file system mounted as *pathname* in the network root directory.

umount: No entry in mtab file for *pathname1*. Deleting gateway object \"*pathname2*\"

You invoked the **/etc/umount** command with the -root option to dismount a file system mounted as *pathname1* in the network root directory. Though the local host does contain the gateway object *pathname2* that corresponds to the mounted file system, the host's /etc/mtab file does not contain an entry for it. The command deleted the gateway object anyway.

umount: *pathname* not in global root. Unmounting anyways

You dismounted a file system mounted in the network root directory, even though the file system did not have the specified mount point *pathname*. However, the mounted file system did have the correct /etc/mtab entry and gateway object before you deleted them with the **/etc/umount** command.

Sometimes mount points disappear from the network root directory because the system administrator has reset the naming server's database to remove obsolete entries. Since mount points do not respond to naming server requests, the naming server deletes mount point entries when rebuilding its database.

umount: umount -root *pathname* must be made from node *//nodename*

You attempted to dismount a file system mounted as *pathname* in the network root directory from the wrong node. (You can dismount a file system only from the same node that mounted it.) Execute the **/etc/umount** command with the -root option on the node *//nodename*.

Usage: mount [-ravpfto [type|option]] ... [fsname] [dir]
[option]=[root,rw,ro,suid,nosuid,quota,noquota,bg,fg,
retry,rsize,wsize,timeo,retrans,port,soft,hard]

Use only the indicated options and arguments with the **/etc/mount** command.

Index

Primary page references are listed first. The letter *f* means "and the following page"; the letters *ff* mean "and the following pages". Symbols are listed at the beginning of the index.

Symbols

/ (slash) 1-5, 2-3
// (slashes) 1-2, 1-5, 2-3f
~ (tilde) 2-11

A

A category site
  definition B-7
  installation procedure B-9f
-a option (mount & umount) A-7
-a option (showmount) A-18
access, unregulated 4-2
accessing
  DOMAIN hosts 1-5, 2-1, 2-7
  file systems 1-2, 1-5, 2-4f
  files with
    AEGIS Shell commands 3-2ff
    Display Manager 3-1f
    DOMAIN/IX commands 3-2
    programs 4-1ff
  objects in mounted file systems 1-4f
adding new users and groups B-14

advantages of DOMAIN NFS
gateway object 1–2
initial mount 1–2
network root directory 1–2, 1–5, 2–6, B–3
AEGIS
Shell commands
restricted 3–3
unapplicable 3–3, 3–5
unsupported 3–4
system calls
restrictions 4–11f
support 4–7ff
arguments for
mount command 2–2, A–7f
umount command 2–7, A–7
arranging servers on network B–1ff
authorization to mount file systems A–2

B

B category site
definition B–7
installation procedure B–10ff
background retry A–7
bg argument A–7
BSD4.2
C library 4–2ff
man command A–1
buffer size
read request A–8
write request A–8
byte seeks 4–14

C

C category site
definition B–7
installation procedure B–12ff
C language
calls 4–2
libraries 4–2ff
-c option A–14
cartridge tape, mounting and dismounting A–6
casehm files 2–13, C–3
categories of sites B–6ff
client
host 1–3f, A–16
statistics A–14
commands
AEGIS Shell
restricted 3–3
unapplicable 3–5
unsupported 3–4
crpasswd B–12, B–14
DOMAIN/IX 3–2
flush_cache B–14
ftp B–10, B–11, B–14
ls C–4
man 1–3, 1–4, A–1
mount 1–3, 2–2, A–4, A–11
nfsstat 1–4, 2–9, A–14
passwd B–10
rpcinfo 1–4, 2–9, A–17
showmount 1–4, 2–9, A–10, A–18
umount 1–4, 2–7, A–4, A–11
compilers, restrictions 3–3
components of NFS software 1–3
configuration,
multiple servers B–5

single server B–3ff
connection flags 4–12
conventions for naming mount points 2–6
creating
processes 3–2
remote files 4–14
crpasswd command B–12, B–14

D

-d option A–18
daemon, server 1–3
definitions
DOMAIN NFS 1–1
foreign
network 1–2
system 1–1
gateway object 1–2
host
client 1–3
server 1–3
mount point 1–2
remote file system 1–2
server 1–3
type manager 1–3
deleting gateway objects 2–8
directory (remote)
access from programs 4–1ff
mounting 1–4
type
identifier 1–3, 4–11
manager 1–3
directories
/ (slash) 1–5
/etc B–11, B–13

/node_data 2–4, 2–5, A–10, A–13, A–16, C–3
/sys/mgrs 1–3
working 1–5
disadvantages of other NFS implementations 1–2, 1–5, 2–6
diskette, mounting and dismounting A–6
dismounting a file system 1–4, 2–7, A–6
Display Manager
compatibility 3–1f
editing invisible files C–2
display of
clients mounted on local host A–18
exportable file systems A–18
message during mount A–8
mounted file systems 2–2, A–6, A–8
NFS statistics 2–9f, A–14
DM, see Display Manager
DOMAIN
C language 4–1ff
FORTRAN language 4–1, 4–13f
host access 1–5, 2–1
language support 4–1ff
network root directory 1–2, 1–5, 2–3ff, B–3
dismounting from 2–8
global version 2–4, 2–8
local copy 2–4, 2–8
mount point 2–7
Pascal language 4–12f
pathnames 2–10
NFS, see NFS
DOMAIN/IX
compatibility 3–2
system
administration node B–11, B–13
calls 4–1ff

E

-e option A-18
editing invisible files C-2
environment variables in pathnames 2-11
error messages D-1ff
establishing uniform IDs on networks B-6ff
/etc directory B-11, B-13
/etc/exports, see exports file
/etc/fstab, see fstab file
/etc/group, see group file
/etc/hosts, see hosts file
/etc/mount, see mount command
/etc/mountd, see mountd server daemon
/etc/mtab, see mtab file
/etc/nfsd, see nfsd server daemon
/etc/nfsstat, see nfsstat command
/etc/passwd, see passwd file
/etc/passwd.map, see passwd.map file
/etc/portmap, see portmap server daemon
/etc/rpcinfo, see rpcinfo command
/etc/showmount, see showmount command
/etc/umount, see umount command
exported file system, definition 2-9
exports file 2-9, A-2, A-10, A-18, C-2
extended pathnames 2-10, 2-13

F

-f option A-7
faking an mtab entry A-7
fg argument A-7
file (remote) 1-3
  access
    programming techniques 4-1
  creation 4-14

  system,
    access authorization A-2
    accessing 1-5, 2-4f
    dismounting 1-4, 2-7, A-6
    displaying mounted 2-2
    multiple 1-5, B-1f
    mounting 2-2ff
    name sharing 1-2, 2-4, B-4
    removable 2-2, 2-7, A-6, A-9
    types A-8
    write-protected A-9
  type
    description 2-11f
    identifier 1-3, 4-11
    manager 1-3, 2-11f
files
  casehm 2-13, C-3
  exports 2-9, A-2, A-10, A-18, C-2
  invisible C-2
  mtab 2-2, A-11
  `node_data/nfs_error_log C-5
  structured 2-12
  unstructured (UASC) 2-12
flags,
  connection 4-12
  object 4-12
flush_cache command B-14
foreground retry A-7
foreign
  file
    access through programs 4-1ff
    limitations 2-12
    system mounting 2-3ff
  host, see system
  network definition 1-2

system
definition 1–1
mounting DOMAIN network 1–5, 2–6, B–3
pathnames 2–10ff
FORTRAN language
compiler 3–3
I/O statements 4–1, 4–13f
fstab file A–4, A–6, A–8
ftp command B–10, B–11, B–14
function of NFS 1–1
functions, C library 4–2ff

G

gateway
node 1–2
object
definition 1–2
deletion 2–8
function 1–4f, 2–2f
gwname argument 2–2ff, 2–6, 2–7
location 2–5
multiple 1–5
suffixes 2–4, 2–5
type identifier 1–3
type manager 1–3, 2–11f
global network root directory
description 2–4, 2–8
dismounting from A–8
mounting in A–8
group
addition to network B–14f
file B–7, B–9f
ID B–6ff
gwname argument 2–2ff, 2–6, 2–7

H

hard
argument 2–3, A–7
mount A–7
host
client 1–3, 1–4
definition 1–3
server 1–3, 1–4, 2–6, B–1ff
host:fsname argument 2–2f, 2–7
hosts file A–2

I

ID, see identifier
identifier
group B–6ff
user B–6ff
initializing statistics A–15
installation procedures
category A site B–9f
category B site B–10ff
category C site B–12ff
installing NFS software 1–3, B–1ff
invocation of server daemons A–10, A–13, A–16, C–2
I/O procedures, standard 4–1, 4–12ff
ios_$ system calls 4–2, 4–7f, 4–11f, 4–14
ios_$create 4–11, 4–14
ios_$inq_rec_pos 4–14
ios_$get 4–14
ios_$set_conn_flag restrictions 4–12
ios_$set_obj_flag restrictions 4–12
ios_$set_rec_type restrictions 4–12

L

language
  compiler restrictions 3–3
  libraries 4–2ff
  standard I/O procedures 4–1ff
limitations
  AEGIS Shell commands 3–3ff
  Display Manager (DM) 3–1f
  DOMAIN/IX commands 3–2
  NFS protocol 2–11f
  other NFS implementations 1–2, 1–5, 2–6, 2–10
  super-user 2–13
links containing special symbols 2–11
listing mounted file systems, see display of
local
  copy of network root directory 2–4, 2–8
  file system 1–4
location of gateway objects 2–5
ls command C–4

M

magnetic tape, see magtape
magtape
  dismounting A–6
  mounting A–6, A–9
man
  command 1–4, A–1
  pages 1–3, 1–4, A–1ff
manager, type 1–3, 2–11f
managing processes 3–2
"manual"
  dismounting from network root directory 2–8f
  mounting in network root directory 2–4f
mknod function 4–7

mount
  command
    description 1–3, 2–2f, A–7ff
    error messages D–1ff
    fstab file A–4
    gwname argument 2–2ff, 2–6, 2–7
    host:fsname argument 2–2, 2–7
    mtab file A–11
    options 2–2ff
    retries A–8
    syntax 2–2, A–7
  point
    definition 1–2
    network root directory 2–7
  request
    management A–10
    service A–13
mountd server daemon 1–3, A–2, A–10, A–18
mounted file system
  display 2–2
  entries
    fstab file A–4
    mtab file A–11
  pathnames 1–4f, 2–4
mountable file systems A–2
mounting
  authorization A–2
  DOMAIN
    host 2–1
    network 1–5, 2–6
  file system 1–4, 2–2ff, A–6ff
  magtape A–9
  management A–10
  multiple file systems 1–5
  network root directory 2–6, B–3
mtab file 2–2, A–7, A–10

multiple
file systems 1–5, B–1f
gateway objects 1–5
servers per network B–5

N

–n option A–14
name_$ system calls 4–9f, 4–14
naming
conventions for mount points 2–6
server helper
not running on the network 2–4f, 2–8
running on the network 2–4, 2–8
NFS
advantages
gateway object 1–2
initial mount 1–2
network root directory 1–2, 1–5, 2–6
components 1–3
definition 1–1
function 1–1
implementation 1–1
installation 1–3, B–1ff
operation 1–4ff
prerequisites 1–2
protocol limitations 2–11f
server 1–3, B–1ff
status display 2–9f, A–14
timeout A–8
troubleshooting C–1ff
type manager 1–3, 2–11f
network
registry B–11, B–15
root directory, see DOMAIN network root directory

Network File System, see NFS
.nfs suffix 2–5
nfs_dir type manager 1–3
nfs_file type manager 1–3
nfs_gate type manager 1–3
.nfs_root suffix 2–4
nfsd server daemon 1–3, A–13
nfsstat command 1–4, A–14
node, gateway 1–2
`node_data directory 2–4, 2–5, A–10, A–13, A–16, C–3
`node_data/nfs_error_log C–5
NS_HELPER, see naming server helper

O

–o option A–7
object
access in mounted file system 1–4f
flag 4–12
gateway 1–4f, 2–2f
online man pages 1–3, 1–4, A–1
operation of NFS software 1–4ff
options
defaults (mount/umount) A–7
mount command 2–2f, A–7f
–o hard 2–3
–o root 2–4
–o soft 2–3
–root 2–8
recommended 2–3
separation 2–4
showmount command 2–9, A–10, A–18
umount command 2–7, A–7
other NFS implementations
disadvantages 1–2, 1–5, 2–6

explicit mounting 1-2, B-3
mount point directory 1-2
remounting 1-2
limitations 1-2, 1-5, 2-6, 2-10

P

-p option A-8
pages, man 1-3, 1-4, A-1
Pascal language
compiler 3-3
I/O procedures 4-1, 4-12f
passwd
command B-10
file B-7, B-10f
passwd.map file B-12
pathname
extension 2-13
for object in mounted file system 1-4f, 2-3f
relativity 2-10
restrictions 2-10
wildcards 2-10
performance of programs 4-14
port
argument A-8
number A-8, A-16
portmap server daemon 1-3, A-16
restarting A-16
prerequisites for DOMAIN NFS 1-2f
probing RPC servers A-17
procedures, standard I/O 4-1, 4-12ff
process
creation 3-2
management 3-2
program

access to files 4-1ff
number mapping A-16
performance 4-14

R

-r option (mount) A-8
-r option (nfsstat) A-14
read request buffer size A-8
read-only
argument A-8
option A-8
read/write A-8
recommended options for mount command 2-3
record seeks 4-14
registry, network B-11, B-15
relative pathnames, 2-10
remote
directory 1-3
file 1-3
file system
definition 1-2
dismounting 1-4, 2-7, A-6, A-11
mounting 2-3ff, A-6, A-11
host
access 2-1
mounting 2-1
removable file system
dismounting 2-7, A-6, A-11
mounting 2-2, A-6, A-9, A-11
restricted AEGIS Shell commands 3-3
restrictions on
AEGIS system calls 4-11f
compilers 3-3
dismounting file systems 2-8

DOMAIN and DOMAIN/IX calls 4–7
pathnames 2–10
retries, mount A–8
retry argument A–8
ro argument A–8
root argument (mount) 2–4, A–8
-root option (umount) 2–8, A–8
RPC (Remote Procedure Call)
data A–14
probing A–17
program number mapping A–16
rpcinfo command 1–4, 2–9, A–17
rsiz argument A–8
rw argument A–8

S

-s option (nfsstat) A–14
-s option (server) A–10, A–13, A–16
seeks, byte and record 4–14
separating options 2–4
server
arrangement on network B–1ff
daemons 1–3, A–10, A–13, A–16, C–2
helper (naming) 2–4f, 2–8f
host 1–3, 1–4, 2–6
option (nfsstat) A–14
statistics A–14
server_host argument 2–9
servicing mount requests A–13
setting
port number A–8
working directory 1–5
sharing file system names 1–2, 2–4, B–4
shell commands
AEGIS 3–2ff
DOMAIN/IX 3–2
showing of mounted file systems, see display of
showmount command 1–4, A–10
description 2–9, A–18ff
options 2–9, A–18
server_host argument 2–9
syntax 2–9, A–18
single NFS server on network B–3ff
site categories B–6f
sleep option A–10, A–13
soft argument 2–3, A–8
software release required by NFS 1–2
special
characters in command lines 3–3
symbols in links 2–11
standard I/O procedures 4–1, 4–12ff
statements, FORTRAN I/O 4–1, 4–13f
statistics (NFS) 2–9f
display A–14
stream_$ system calls 4–7, 4–10ff
stream_$redefine restrictions 4–12
structured file limitations 2–12
suffixes
.nfs 2–5
.nfs_root 2–4
super-user limitation 2–13
syntax
mount 2–2, A–6
nfsstat A–14
rpcinfo A–17
showmount 2–9, A–18
umount 2–7, A–6
System V
C library 4–2ff
man command A–1

/sys/mgrs directory 1–3
system administration node B–11, B–13
system calls
  AEGIS
    ios_$ 4–2, 4–7f, 4–11f, 4–14
    name_$ 4–9f, 4–14
    restrictions 4–11f
    stream_$ 4–7, 4–10ff
  DOMAIN/IX 4–1ff

T

-t option A–8
TCP/IP software required by NFS 1–2
timeo argument A–8
timeout A–8
troubleshooting NFS C–1ff
type
  identifiers 1–3, 4–11
  manager
    components 1–3
    definition 1–3, 2–11
    function 2–11f
types, file system A–8

U

unstructured (UASC) files 2–12
umount command 1–4
  description 2–7, A–6
  error messages D–3f
  fstab file A–4
  host:fsname argument 2–7
  gwname argument 2–7
  mtab file A–11

  options 2–7
  restriction 2–8
  syntax 2–7, A–6
unmounting, see dismounting
unregulated access 4–2
unsupported AEGIS Shell commands 3–4
user
  addition to network B–14f
  ID B–6ff
utility commands, see commands

V

-v option A–8

W

wildcards 2–10, 3–2
working directory setting 1–5
write request buffer size A–8
write-protected file systems A–9
wsiz argument A–8

Z

-z option A–15
zero statistics A–15